

**SIMULASI PEMANFAATAN PHP XML SEBAGAI WEB SERVICE
MENGUNAKAN TEKNOLOGI CLOUD COMPUTING
DENGAN METODE SOA UNTUK MENINGKATKAN
KINERJA SISTEM REGISTRASI
MAHASISWA UNIB**

SKRIPSI



Oleh :

ROZY FETRISIA

G1A009032

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS BENGKULU

2014

**SIMULASI PEMANFAATAN PHP XML SEBAGAI WEB SERVICE
MENGUNAKAN TEKNOLOGI CLOUD COMPUTING
DENGAN METODE SOA UNTUK MENINGKATKAN
KINERJA SISTEM REGISTRASI
MAHASISWA UNIB**

SKRIPSI

**Diajukan Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana pada Fakultas
Teknik Program Studi Teknik Informatika
Universitas Bengkulu**



Oleh :

ROZY FETRISIA

G1A009032

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS BENGKULU**

2014

MOTTO

“Karya yang sederhana tapi mempunyai kualitas yang luar biasa”

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

- ✓ **Allah SWT.**
- ✓ **Kitab Suci Al-Qur'an.**
- ✓ **Nabi Muhammad SAW.**
- ✓ **Kedua Orang Tua yang selalu aku do'akan dan aku banggakan sampai akhir hayat Ku (Supratman dan Ermawaty), terima kasih karena kalian selalu mendukung, memotivasi dan mendoakan dalam kesuksesan Ku selama ini.**
- ✓ **Pengorbanan kalian sangat berarti dan tidak bisa dibayar dengan apapun.**
- ✓ **Yang Tersayang (Retno Wahyu Ningsih) yang selalu menyemangatiku.**
- ✓ **Sahabat Informatika 2009 : Abdur-Abner-Apni-Bobby-Dian-Dita-Disa-Eko-Firdaus-Fuad-Gita-Gufron-Handri-Irawan-Julia-Leni-Lian-Linda-Mei-Odi-Radiaz-Randi-Rewa-Rinov-Rofika-Ryza-Yodi-dan lainnya.**
- ✓ **Sahabatku : Ayu-Ade-Ari-Dedy-Etang-Fitri-Hery-Jesika-Ibnu-Iqbal-Ihsan-Ilma-Mulya-Ogi-Resty.**
- ✓ **Almamaterku**

KATA PENGANTAR



Assalamu'alaikum Wr.Wb.

Alhamdulillahirabbil'alamin, puji syukur penulis ucapkan kepada Allah SWT yang telah memberikan rahmat serta hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Simulasi Pemanfaatan Php Xml Sebagai Web Service Menggunakan Teknologi Cloud Computing dengan Metode Soa Untuk Meningkatkan Kinerja Sistem Registrasi Mahasiswa Unib”. Penulisan skripsi ini disusun sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Teknik di Program Studi Teknik Informatika Universitas Bengkulu.

Selesainya penyusunan skripsi ini tidak terlepas dari bimbingan, arahan, masukan dan bantuan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Bapak Khairul Amri, S.T, M.T sebagai Dekan Fakultas Teknik Universitas Bengkulu
2. Ibu Desi Andreswari, S.T., M.Cs sebagai Ketua Program Studi Teknik Informatika yang telah memberikan kesempatan kepada saya untuk menyelesaikan skripsi ini
3. Bapak Aan Erlanshari, S.T., M.Eng sebagai dosen pembimbing utama yang telah meluangkan waktu memberikan bimbingan dan motivasi dalam menyelesaikan skripsi ini.

4. Bapak Funny Farady Coastera S.Kom, M.T sebagai dosen pembimbing pendamping yang telah banyak sabar menghadapi penulis dalam bimbingan skripsi ini
5. Ibu Dr. Diah Puspitaningrum, S.T., M.Kom dan Bapak Rusdi Efendi, S.T., M.Kom sebagai dosen penguji utama dan penguji pendamping yang telah memberikan masukan-masukan demi penyempurnaan skripsi ini.
6. Seluruh dosen dan staf karyawan, serta seluruh civitas akademika Program Studi
7. Ibu, ayah dan adik-adikku yang telah mendoakan, memberikan dukungan dan memotivasi dalam menyelesaikan skripsi ini.
8. Teman-teman seperjuangan Teknik Informatika Angkatan 2009 dan 2010.
9. Semua pihak yang sudah membantu dan tidak dapat penulis sebutkan satu persatu.

Penulis berharap semoga Allah SWT mengaruniakan rahmat dan hidayah-Nya kepada mereka semua. Semoga skripsi ini dapat bermanfaat bagi kita semua, *aamiin*.

Wassalamu'alaikum Wr. Wb.

Bengkulu, 23 Juni 2014

Penulis

**SIMULASI PEMANFAATAN PHP XML SEBAGAI WEB SERVICE
MENGUNAKAN TEKNOLOGI CLOUD COMPUTING
DENGAN METODE SOA UNTUK MENINGKATKAN
KINERJA SISTEM REGISTRASI
MAHASISWA UNIB**

Oleh
Rozy Fetrisia
G1A009032

Abstrak

Penelitian ini bertujuan untuk merancang dan membangun suatu simulasi *webservice* dengan PHP XML yang menerapkan *Service Oriented Architecture* pada interkoneksi antara server akademik dan *server* Bank. Sistem ini diharapkan dapat memberikan solusi permasalahan interkoneksi antar *server* akademik dengan *server* perbankan, memudahkan pihak Bank untuk dapat memberikan informasi mahasiswa yang telah melakukan proses pembayaran akademik secara *realtime* kepada pihak UNIB serta memudahkan pihak UNIB dalam memperoleh informasi mahasiswa yang telah melakukan proses pembayaran akademik dari pihak Bank secara *realtime*. Aplikasi ini dibangun dengan menggunakan bahasa pemrograman PHP XML dengan Adobe Dreamweaver CS4. Metode pengembangan sistem yang digunakan untuk membangun aplikasi ini adalah model *Unified Modeling Language* (UML) sebagai perancangan sistem. Hasil akhir dari penelitian ini adalah terciptanya sebuah *webservice* yang dapat memberikan informasi mahasiswa yang telah melakukan proses pembayaran akademik dari pihak Bank kepada pihak UNIB secara *realtime*.

Kata kunci: PHP XML, Web Service, *Cloud Computing*, *Service Oriented Architecture*, UNIB

***SIMULATION USING PHP XML WEB SERVICE USING TECHNOLOGY CLOUD
COMPUTING WITH SOA METHOD FOR IMPROVING
STUDENT REGISTRATION SYSTEM PERFORMANCE OF
UNIB***

*By
Rozy Fetrisia
G1A009032*

Abstract

The aim of this research is to design and build a simulation PHP XML web service with Service Oriented Architecture that implements the interconnection between academic server and banking server. This system is expect to provide solutions to academic server interconnection between banking server with academic server, enabling the bank to be able to provide information that the student has academic payments in real time to UNIB and facilitate in obtaining the information that the student has made the payment of bank`s academic in real time. This application was created using the PHP programming language XML with Adobe Dreamweaver CS4. The method used in the development of the program is unified modeling language (UML) as the system design. The result of this research is the creation of webservice that can provide students who have made the process academic payment from the Bank to the UNIB in realtime.

Keywords: *PHP XML, Web Service, Cloud Computing, Service Oriented Architecture, UNIB*

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN.....	iii
MOTTO DAN PERSEMBAHAN	iv
KATA PENGANTAR.....	v
ABSTRAK	vi
<i>ABSTRACT</i>.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xi
DAFTAR TABEL	xiv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	3
1.5. Manfaat Peneltian.....	4
1.6. Sistematika Penulisan.....	4
BAB II LANDASAN TEORI.....	6
2.1. Universitas Bengkulu	6
2.1.1 Sejarah.....	6
2.1.2 Visi Misi.....	7
2.2. <i>eXtensible Markup Language</i>	8
2.3. Teknologi <i>Cloud</i>	10
2.3.1 Model Layanan <i>Cloud Computing</i>	11
2.3.2 Model <i>Cloud Computing</i>	13
2.4. <i>Service Oriented Architecture</i>	14
2.5. Metode Pengembangan Sistem	15
2.6. <i>Unified Modeling Language</i>	18
2.7. Pengujian Sistem	24
2.7.1 Teknik Pengujian Sitem	24
2.7.2 Pengukuran Tingkat Kepuasan Pengguna	25
2.8. Penelitian Terkait	26
BAB III METODE PENELITIAN	29
3.1. Jenis Penelitian	29
3.2. Sarana Pendukung	29
3.3. Jenis dan Sumber Data	30
3.4. Teknik Pengumpulan Data	30

3.5. Metode Pengembangan Sistem	31
3.6. Metode Uji Kelayakan Sistem	34
3.7. Jadwal Penelitian.....	38
BAB IV ANALISIS DAN DESAIN PERANGKAT LUNAK	39
4.1. Analisa Sistem	39
4.1.1. Analisis Permasalahan	39
4.1.2. Analisis Fungsional.....	40
4.1.3. Analisis Non Fungsional.....	40
4.1.4. Alur Sistem Yang Dibangun	41
4.1.4.1 .Alur Sistem Unib.....	41
4.1.4.2 .Alur Sistem Bank.....	46
4.2. Perancangan Sistem	48
4.2.1. Perancangan UML	48
4.2.2. Perancangan Antar Muka (<i>User Interface</i>)	56
4.2.2.1 .Perancangan Antar Muka Pihak Unib.....	57
4.2.2.2 .Perancangan Antar Muka Pihak Bank.....	77
4.2.3 .Perancangan Basis Data.....	82
4.2.3.1 .Struktur Basis Data.....	82
BAB V HASIL DAN PEMBAHASAN	94
5.1. Implementasi Sistem	94
5.2. Pengujian Sistem	97
5.2.1 Pengujian <i>White-Box</i>	97
5.2.2 Pengujian <i>Black-Box</i>	102
5.2.3 Uji Kelayakan Sistem.....	109
BAB VI KESIMPULAN DAN SARAN	114
6.1. Kesimpulan.....	114
6.2. Saran	114
DAFTAR LAMPIRAN.....	xv
DAFTAR PUSTAKA.....	
LAMPIRAN.....	

DAFTAR GAMBAR

Gambar 2.1 Tampilan dokumen XML pada browser	10
Gambar 2.2 <i>Cloud Computing</i>	11
Gambar 2.3 Metode <i>Waterfall</i>	15
Gambar 2.4 <i>Three complementary views or sets of diagrams</i>	20
Gambar 2.5 <i>Class diagram</i>	20
Gambar 2.6 <i>Activitydiagram</i>	21
Gambar 2.7 <i>Usecase diagram</i>	22
Gambar 2.8 <i>Sequance diagram</i>	23
Gambar 3.1 Diagram alir penelitian.....	31
Gambar 4.1 Diagram Alir Admin UNIB	42
Gambar 4.2 Diagram Alir Operator UNIB	45
Gambar 4.3 Diagram Alir Bank.....	47
Gambar 4.4 <i>Use Case Diagram</i>	49
Gambar 4.5 <i>Activity Diagram</i>	50
Gambar 4.6 <i>Sequence Diagram</i>	52
Gambar 4.7 <i>Class Diagram</i> Unib	54
Gambar 4.8 <i>Class Diagram</i> Bank.....	56
Gambar 4.9 <i>Form</i> Menu Login Unib.....	57
Gambar 4.10 <i>Form Login</i> Admin	58
Gambar 4.11 <i>Form</i> Menu Admin	58
Gambar 4.12 <i>Form</i> Menu Master Mahasiswa	59
Gambar 4.13 <i>Form</i> Menu Tambah Master Mahasiswa	59
Gambar 4.14 <i>Form</i> Menu Edit Master Mahasiswa.....	60
Gambar 4.15 <i>Form</i> Menu Hapus Master Mahasiswa	60
Gambar 4.16 <i>Form</i> Biaya Potongan Mahasiswa	61
Gambar 4.17 <i>Form</i> Menu Tambah Biaya Potongan Mahasiswa	61
Gambar 4.18 <i>Form</i> Menu Edit Biaya Potongan Mahasiswa	62
Gambar 4.19 <i>Form</i> Menu Hapus Biaya Potongan Mahasiswa.....	62
Gambar 4.20 <i>Form</i> Menu Admin	63
Gambar 4.21 <i>Form</i> Menu Tambah Admin	63
Gambar 4.22 <i>Form</i> Menu Edit Admin.....	64
Gambar 4.23 <i>Form</i> Menu Hapus Admin	64
Gambar 4.24 <i>Form</i> Menu Operator	65
Gambar 4.25 <i>Form</i> Menu Tambah Operator	65
Gambar 4.26 <i>Form</i> Menu Edit Operator.....	66
Gambar 4.27 <i>Form</i> Menu Hapus Operator	66
Gambar 4.28 <i>Form</i> Menu Bank	67
Gambar 4.29 <i>Form</i> Menu Tambah Bank.....	67
Gambar 4.30 <i>Form</i> Menu Edit Bank	68
Gambar 4.31 <i>Form</i> Menu Hapus Bank.....	68
Gambar 4.32 <i>Form</i> Menu Biaya	69
Gambar 4.33 <i>Form</i> Menu Tambah Biaya.....	69

Gambar 4.34 <i>Form</i> Menu Edit Biaya	70
Gambar 4.35 <i>Form</i> Menu Hapus Biaya.....	70
Gambar 4.36 <i>Form</i> Menu Rincian Biaya.....	71
Gambar 4.37 <i>Form</i> Menu Tambah Rincian Biaya.....	71
Gambar 4.38 <i>Form</i> Menu Edit Rincian Biaya	72
Gambar 4.39 <i>Form</i> Menu Hapus Rincian Biaya	72
Gambar 4.40 <i>Form</i> Menu Prodi.....	73
Gambar 4.41 <i>Form</i> Menu Tambah Prodi.....	73
Gambar 4.42 <i>Form</i> Menu Edit Prodi	74
Gambar 4.43 <i>Form</i> Menu Hapus Prodi.....	74
Gambar 4.44 <i>Form</i> Menu Fakultas	75
Gambar 4.45 <i>Form</i> Menu Tambah Fakultas.....	75
Gambar 4.46 <i>Form</i> Menu Edit Fakultas	76
Gambar 4.47 <i>Form</i> Menu Hapus Fakultas.....	76
Gambar 4.48 <i>Form</i> Menu Laporan	77
Gambar 4.49 <i>Form</i> Menu <i>Login</i> Operator	77
Gambar 4.50 <i>Form</i> Menu <i>Login</i> Admin Bank.....	78
Gambar 4.51 <i>Form</i> Menu Utama Admin Bank	78
Gambar 4.52 <i>Form</i> Menu Admin	79
Gambar 4.53 <i>Form</i> Menu Operator	79
Gambar 4.54 <i>Form</i> Menu Pembayaran.....	80
Gambar 4.55 <i>Form</i> Menu Profil Admin	80
Gambar 4.56 <i>Form</i> Menu Login Operator.....	80
Gambar 4.57 <i>Form</i> Menu Utama Operator Bank	81
Gambar 4.58 <i>Form</i> Menu Pembayaran.....	81
Gambar 4.59 <i>Form</i> Menu Ubah Profil Operator	82
Gambar 4.60 <i>Entity Relation Diagram</i> Unib	91
Gambar 4.61 <i>Entity Relation Diagram</i> Bank.....	92
Gambar 5.1 Grafik Persentase Hasil Angket Variabel 1	111
Gambar 5.2 Grafik Persentase Hasil Angket Variabel 2	112
Gambar 5.3 Grafik Persentase Hasil Angket Variabel 3	113

DAFTAR TABEL

Tabel 3.1 Jadwal Penelitian yang akan dilaksanakan	38
Tabel 4.1 Kelas, Atribut dan Metode Unib.....	54
Tabel 4.2 Kelas, Atribut dan Metode Bank	56
Tabel 4.3 Tabel Admin	83
Tabel 4.4 Tabel Bank.....	83
Tabel 4.5 Tabel Fakultas.....	84
Tabel 4.6 Tabel Mahasiswa	84
Tabel 4.7 Tabel Operator	85
Tabel 4.8 Tabel Pembayaran Mahasiswa.....	86
Tabel 4.9 Tabel Prodi.....	86
Tabel 4.10 Tabel Rincian Biaya.....	87
Tabel 4.11 Tabel Total Biaya.....	87
Tabel 4.12 Tabel Admin Bank.....	88
Tabel 4.13 Tabel Operator Bank.....	89
Tabel 4.14 Tabel Pembayaran.....	89
Tabel 5.1 Daftar <i>Layout</i> Aplikasi.....	93
Tabel 5.2 Daftar Modular Aplikasi.....	94
Tabel 5.3 Pengujian <i>Black Box</i>	101
Tabel 5.4 Kategori Penilaian.....	110
Tabel 5.5 Hasil Penilaian Variabel Tampilan	111
Tabel 5.6 Hasil Penilaian Variabel Kemudahan Pengguna	112
Tabel 5.7 Hasil Penilaian Variabel Kinerja Sistem	113

DAFTAR LAMPIRAN

Lampiran A Surat Izin Penelitian	A-1
Lampiran B Sample Data Mahasiswa.....	B-1
Lampiran C Pengujian <i>Black -Box</i>	D-1
Lampiran D Formulir Angket Bank	D-1
Lampiran D Formulir Angket Unib.	D-2
Lampiran E Tabulasi Data Hasil Perhitungan Angket Uji Kelayakan	E-1

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era yang semakin maju ini, kebutuhan seseorang untuk sarana informasi sangatlah penting. Dengan adanya teknologi informasi ini, dapat memudahkan seseorang dalam memperoleh sebuah informasi, salah satunya adalah informasi pembayaran. Ditinjau dari keadaan saat ini, pegawai unib perlu menunggu untuk dapat melihat data-data mahasiswa yang telah melakukan pembayaran. Dengan adanya sebuah teknologi informasi membuat pegawai unib dapat memperoleh informasi data-data orang yang telah melakukan pembayaran tanpa perlu menunggu untuk mendapatkan informasi tersebut.

Sistem pembayaran uang kuliah di Universitas Bengkulu yang dilakukan melalui teller pada bank BNI merupakan salah satu cara untuk mempermudah mahasiswa dalam melakukan pembayaran uang kuliah. Cara pembayaran ini secara umum dapat dikatakan kompleks, akan tetapi dilihat dari keadaan yang ada pada sampai saat ini data yang ada tidak *realtime* dan masih membutuhkan waktu 1 hari setelah pembayaran untuk dapat melihat daftar mahasiswa yang telah melakukan pembayaran uang kuliah pada bank BNI tersebut.

Kondisi nyata yang ada pada UNIB(Universitas Bengkulu) sekarang ini melayani proses pembayaran dengan *model peer to host* yang tidak *realtime*. Selain itu, karena akses ke *server* Bank tidak mudah maka dibuatlah sebuah simulasi seperti yang ada pada Bank. Proses akses *host to host* antar *server*

juga tidak mudah karena ada proses keamanan pada masing-masing *server* yang tentu saja berbeda-beda. Selain itu juga, mesin server pada masing-masing *server* berbeda-beda *platform* mesinnya.

Dari latar belakang di atas, maka diperlukan sebuah aplikasi untuk pihak bank BNI dan untuk pihak Universitas Bengkulu yang dihubungkan dengan menggunakan metode SOA untuk dapat mengatasi masalah yang ada dan disimulasikan pada server *cloud* yang ada dengan menggunakan teknologi *cloud*, sehingga server UNIB dan server Bank dapat saling terkoneksi dan dapat meningkatkan kinerja sistem registrasi mahasiswa Universitas Bengkulu. Oleh karena itu, dilakukanlah penelitian dan memilih judul yaitu **“Simulasi Pemanfaatan Php Xml Sebagai Web Service Menggunakan Teknologi Cloud Computing Dengan Metode SOA Untuk Meningkatkan Kinerja Sistem Registrasi Mahasiswa Unib”**. Sehingga dengan adanya simulasi ini, diharapkan dapat membuat bagian akademik untuk memonitor dan mengolah registrasi mahasiswa ke Bank.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah yang akan dibahas pada penelitian ini adalah Bagaimana mensimulasikan penerapan *Service Oriented Architecture* sebagai *web service* untuk mengatasi masalah interkoneksi antara *server* perbankan dengan *server* Universitas Bengkulu dengan PHP XML pada teknologi cloud untuk meningkatkan kinerja sistem registrasi mahasiswa Universitas Bengkulu ?

1.3 Batasan Masalah

Dalam menerapkan *Service Oriented Architecture* pada Interkoneksi antara server akademik dan server bank, ditetapkan berbagai batasan masalah sebagai berikut :

1. Proses bisnis perbankan yang disimulasikan ini hanya dalam transaksi pembayaran akademik.
2. Data yang akan digunakan sebagai data uji pada Simulasi ini merupakan format data yang didapatkan dari pihak Bank BNI dan sample data akademik mahasiswa yang melakukan proses pembayaran mahasiswa baru, registrasi ulang.
3. Simulasi yang akan di bangun ini menggunakan bahasa pemrograman PHP dan dengan menggunakan server cloud yang sudah ada seperti cPanel.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dari penelitian ini adalah merancang dan membangun suatu simulasi PHP XML yang menerapkan *Service Oriented Architecture* pada interkoneksi antara server akademik dan server Bank adalah :

1. Memberikan solusi permasalahan interkoneksi antar server akademik dengan server perbankan.
2. Memudahkan pihak Bank untuk dapat memberikan informasi mahasiswa yang telah melakukan proses pembayaran akademik secara *realtime* kepada pihak UNIB.

3. Memudahkan pihak UNIB dalam memperoleh informasi mahasiswa yang telah melakukan proses pembayaran akademik dari pihak Bank secara *realtime*

1.5 Manfaat Penelitian

Adapun manfaat yang dicapai dari penelitian ini adalah :

1. Manfaat bagi pihak Bank adalah dapat memberikan informasi mahasiswa yang telah membayar uang kuliah secara cepat atau *realtime*
2. Manfaat bagi pihak UNIB adalah dapat memperoleh informasi mahasiswa yang telah melakukan pembayaran dari pihak Bank tanpa harus menunggu 1 hari setelah melakukan proses pembayaran

1.6 Sistematika Penulisan

Dalam penyusunan tugas akhir ini, sistematika penulisan sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisikan pembahasan masalah umum yang meliputi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini memuat landasan teori yang berfungsi sebagai sumber atau referensi dalam memahami permasalahan yang berkaitan dengan aplikasi yang akan dibangun.

BAB III METODE PENELITIAN

Bab ini berisi tentang metode-metode yang digunakan dalam penelitian, seperti jenis penelitian, teknik pengumpulan data, jenis dan sumber data, metode pengembangan sistem, metode pengujian dan jadwal penelitian..

BAB IV ANALISIS DAN DESAIN PERANGKAT LUNAK

Bab ini menjelaskan setiap tahapan analisis dan perancangan sistem aplikasi yang akan dibangun dalam penelitian meliputi analisis sistem dan perancangan sistem.

BAB V HASIL DAN PEMBAHASAN

Bab ini merupakan bab yang berisi hasil dan pembahasan yang menguraikan hasil perancangan sistem dan implementasinya..

BAB VI PENUTUP

ini merupakan bab penutup yang merupakan bab terakhir yang berisi kesimpulan dan saran dari pembuatan tugas akhir sampai ke pengembangan perangkat lunak kedepannya.

BAB II

LANDASAN TEORI

2.1 Universitas Bengkulu

Universitas Bengkulu adalah perguruan tinggi negeri yang terletak di Bengkulu, Indonesia, yang berdiri pada 24 April 1982 berdasarkan keputusan Presiden RI Nomor 17 tahun 1982 dan diresmikan oleh Menteri Pendidikan dan Kebudayaan Republik Indonesia, Prof. Dr. Daud Yusuf. UNIB masuk dalam daftar "50 Universitas Terbaik di Indonesia" versi Direktorat Jenderal Pendidikan Tinggi. Kampus UNIB terletak di 3 lokasi berbeda yaitu Kandang Limun (Kompleks Kampus Utama), Cimanuk, dan Air Sebakul((Buku Panduan Akademik, 2007).

2.1.1 Sejarah

Universitas Bengkulu (UNIB) didirikan berdasarkan keputusan Presiden RI Nomor 17 tahun 1982 dan diresmikan oleh Menteri Pendidikan dan Kebudayaan Republik Indonesia, Prof. Dr. Daud Yusuf. Pada saat yang sama dilantik Rektor UNIB pertama, Prof. Ir. Soenjoto Sumodihardjo (UGM) untuk masa jabatan 1982-1986. Dr. Ir. Soekotjo (UGM), yang sebelumnya menjabat Pembantu Rektor I, memimpin UNIB untuk periode 1986-1990. Untuk periode 1990-1995 Dr. Ir. Nitza Arbi (UNAND) diberi kepercayaan memimpin UNIB. Tahun 1995 sampai dengan 2005 jabatan Rektor UNIB dipegang oleh Prof. Dr. H Zulkifli Husin, S.E, M.Sc. (UNSYIAH). Sejak 2005 sampai dengan sekarang Rektor UNIB dijabat oleh Dr. Ridwan Nurazi, S.E., Msc.

Keberadaan UNIB merupakan wujud nyata dari perjuangan yang tak kenal menyerah dari Gubernur Soeprapto yang mendapat dukungan penuh dari masyarakat, tokoh adat, Pemda Tk. I Bengkulu, dan perguruan tinggi swasta bernama Universitas Semarak Bengkulu (UNSEB). Dukungan universitas ini diwujudkan dalam bentuk penyerahan mahasiswa UNSEB sebagai cikal bakal UNIB beserta lahan Kampus seluas 24,9 Ha di Desa Beringin Raya Bengkulu (Buku Panduan Akademik, 2007).

Kendala utama yang dihadapi pada saat proses pendirian UNIB berdasarkan hasil studi kelayakan yang dilakukan oleh Universitas Sriwijaya di antaranya yaitu tidak tersedianya tenaga edukatif, sehingga pada waktu itu disimpulkan bahwa di Bengkulu belum layak didirikan sebuah universitas negeri. Kendala tersebut akhirnya dapat diatasi setelah Gubernur Soeprato menjalin kerjasama dengan Universitas Gadjah Mada Yogyakarta atas petunjuk Presiden Soeharto. Realisasi dari hasil kerjasama tersebut dikirimlah beberapa tenaga edukatif dari UGM antara lain Prof. Ir. Soenjoto Sumodihardjo, Dr. Ir. Soekotjo, Drs. Sutarto, Ir. Supratoyo dan H. Hidjazi, S.H. untuk diperbantukan di UNIB baik sebagai pejabat struktural maupun sebagai tenaga edukatif. Mereka melakukan rekrutmen tenaga edukatif dari beberapa universitas di Jawa dan Sumatera ((Buku Panduan Akademik, 2007).

2.1.2 Visi Misi

a. Visi

Universitas Bengkulu akan menjadi perguruan tinggi yang berada di barisan terdepan dalam pengembangan ilmu pengetahuan, teknologi dan

seni yang mendukung berkelanjutan yang beretika dan bermoral dalam suatu sistem akademik yang demokratis dan didukung oleh fasilitas yang efisien, efektif dan terpadu serta memberikan pelayanan yang memuaskan((Buku Panduan Akademik, 2007).

b. Misi

- Meningkatkan dan mengembangkan program-program akademik unggulan, kemitraan dengan dunia usaha, pemerintah dan masyarakat;
- Meningkatkan suasana akademik yang lebih beretika, bermoral, dan demokratis;
- Melaksanakan pendidikan, penelitian, dan pengabdian kepada masyarakat untuk mencerdaskan dan memenuhi kebutuhan masyarakat;
- Meningkatkan kemampuan kompetisi lulusan Universitas Bengkulu melalui program-program akademik yang strategis, efektif, komprehensif dan relevan;
- Meningkatkan kualitas citivitas akademika Universitas Bengkulu.

2.2 XML

XML merupakan kependekan dari *eXtensible Markup Language*. XML untuk saat ini bukan merupakan pengganti HTML. Masing-masing dikembangkan untuk tujuan yang berbeda. Kalau HTML digunakan untuk menampilkan informasi dan berfokus pada bagaimana informasi terlihat, XML mendeskripsikan susunan informasi dan berfokus pada informasi itu

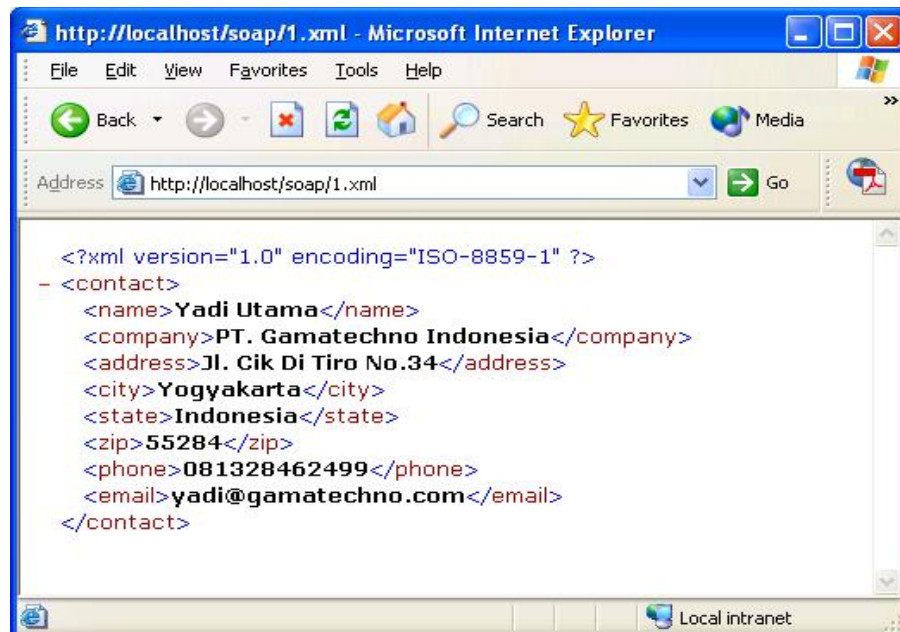
sendiri. XML terutama dibutuhkan untuk menyusun dan menyajikan informasi dengan format yang tidak mengandung format standard layaknya heading, paragraph, table dan lain sebagainya(Akhmad Dharman Kasman, 2013).

Berikut ini adalah contoh sebuah dokumen XML untuk informasi contact person:

```
<?xml version="1.0" encoding="ISO-8859-1"?><contact>
  <contact>
    <name>Yadi Utama</name>
    <company>PT. Gamatechno Indonesia</company>
    <address>Jl. Cik Di Tiro No.34</address>
    <city>Yogyakarta</city>
    <state>Indonesia</state>
    <zip>55284</zip>
    <phone>081328462499</phone>
    <email>yadi@gamatechno.com</email>
  </contact>
```

Seperti halnya HTML, XML juga menggunakan elemen yang ditandai dengan tag pembuka (diawali dengan ‘<’ dan diakhiri dengan ‘>’), tag penutup(diawali dengan ‘</’ diakhiri ‘>’) dan atribut elemen(parameter yang dinyatakan dalam tag pembuka misal <form name=”isidata”>). Hanya bedanya, HTML medefinisikan dari awal tag dan atribut yang dipakai didalamnya, sedangkan pada XML kita bisa menggunakan tag dan atribut sesuai kehendak kita. Berikut ini adalah tampilan dokumen XML di atas

apabila dijalankan pada sebuah browser pada Gambar 2.1(Akhmad Dharman Kasman, 2013):



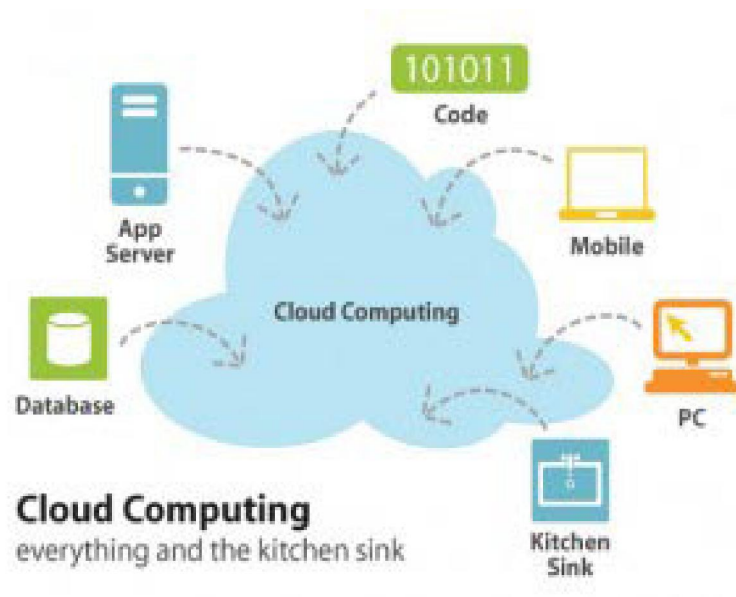
Gambar 2.1: Tampilan dokumen XML pada browser

2.3 Teknologi Cloud

Cloud computing merupakan penggabungan pemanfaatan teknologi komputasi dan pengembangan berbasis internet yang menawarkan fasilitas sharing sumber daya tanpa perangkat tambahan, biaya yang lebih terjangkau, dan penyimpanan data yang tidak terbatas. Dengan menggunakan *Cloud Computing*, aplikasi ini dapat di simpan di awan sehingga dapat diakses oleh pengguna dimanapun pengguna tersebut berada selama pengguna tersebut dapat mengakses internet(Khamidah, Sulistianingsih & Paputungan dalam Wahyudi Tri (2013)).

Cloud atau awan merupakan metafora dari internet, sebagaimana awan yang sering digambarkan pada diagram jaringan komputer. Awan (cloud) dalam cloud computing juga merupakan abstraksi dari infrastruktur kompleks

yang disembunyikannya yaitu suatu moda komputasi dimana kapabilitas terkait teknologi informasi disajikan sebagai suatu layanan (*as a service*), sehingga pengguna dapat mengaksesnya lewat Internet (di dalam awan) tanpa pengetahuan tentangnya, ahli dengannya, atau memiliki kendali terhadap infrastruktur teknologi yang membantunya seperti yang ada pada Gambar 2.2 (Syamsumar & Zen dalam Wahyudi Tri (2013)).



Gambar 2.2: *Cloud Computing*

2.3.1 Model Layanan *Cloud Computing*

Model Layanan *Cloud Computing* di bagi menjadi tiga yaitu diantaranya (Mell dan Grance dalam Syaikh Ahmad (2013)):

1. *Cloud Software as a Service (SaaS)*.

Kemampuan yang diberikan kepada konsumen untuk menggunakan aplikasi penyedia dapat beroperasi pada infrastruktur awan. Aplikasi dapat diakses dari berbagai perangkat klien melalui antarmuka seperti web browser (misalnya, email berbasis web). Konsumen tidak mengelola

atau mengendalikan infrastruktur awan yang mendasari termasuk jaringan, server, sistem operasi, penyimpanan, atau bahkan kemampuan aplikasi individu, dengan kemungkinan pengecualian terbatas terhadap pengaturan konfigurasi aplikasi pengguna tertentu.

2. *Cloud Platform as a Service (PaaS).*

Kemampuan yang diberikan kepada konsumen untuk menyebarkan aplikasi yang dibuat konsumen atau diperoleh ke infrastruktur komputasi awan menggunakan bahasa pemrograman dan peralatan yang didukung oleh provider. Konsumen tidak mengelola atau mengendalikan infrastruktur awan yang mendasari termasuk jaringan, server, sistem operasi, atau penyimpanan, namun memiliki kontrol atas aplikasi disebarkan dan memungkinkan aplikasi melakukan hosting konfigurasi.

3. *Cloud Infrastructure as a Service (IaaS).*

Kemampuan yang diberikan kepada konsumen untuk memproses, menyimpan, berjaringan, dan komputasi sumberdaya lain yang penting, dimana konsumen dapat menyebarkan dan menjalankan perangkat lunak secara bebas dapat mencakup sistem operasi dan aplikasi. Konsumen tidak mengelola atau mengendalikan infrastruktur awan yang mendasari tetapi memiliki kontrol atas sistem operasi, penyimpanan, aplikasi yang disebarkan, dan mungkin kontrol

terbatas komponen jaringan yang pilih (misalnya, firewall host).

2.3.2 Model *Cloud Computing*

Ada empat model *cloud computing* yang di akui yaitu antara lain(Marks & Lozano dalam Wahyudi Tri (2013)):

a. Private Cloud

Infrastruktur cloud yang digunakan hanya untuk organisasi. Infrastruktur ini dapat dikelola oleh organisasi yang bersangkutan ataupun pihak ketiga.

b. Public Cloud

Infrastruktur cloud dibuat dan disediakan untuk masyarakat umum atau kelompok industri besar dan dimiliki oleh sebuah organisasi penjual layanan cloud.

c. Community Cloud

Infrastruktur cloud yang digunakan bersama oleh beberapa organisasi dan mendukung komunitas tertentu. Infrastruktur ini biasanya dikelola oleh organisasi yang bersangkutan atau pihak ketiga.

d. Hybrid Cloud

Kombinasi dari dua atau lebih cloud (private, community, atau public) yang tetap entitas unik namun terikat bersama oleh teknologi standar atau kepemilikan yang memungkinkan portabilitas data dan aplikasi

2.4 SOA (*Service Oriented Architecture*)

Service Oriented Architecture (SOA) adalah prinsip desain yang diinginkan untuk membangun system terdistribusi yang reliable dan mengirimkan service fungsionalitasnya, dengan menambahkan penekanan pada loose coupling antar pengguna service. Dengan kata lain SOA adalah suatu pendekatan arsitektur system yang merepresentasikan proses bisnis menjadi bentuk service yang lebih efisien dan tingkat ketergantungan yang lebih rendah satu sama lain. Dengan menerapkan SOA, aplikasi ini dapat terhubung dengan aplikasi lainnya dan saling mengintegrasikan data satu sama lain. Dengan SOA ini, aplikasi ini dapat terhubung dengan aplikasi lainnya yang berbeda platform mesinnya (Erl dalam Miharjaya Kardy (2013)).

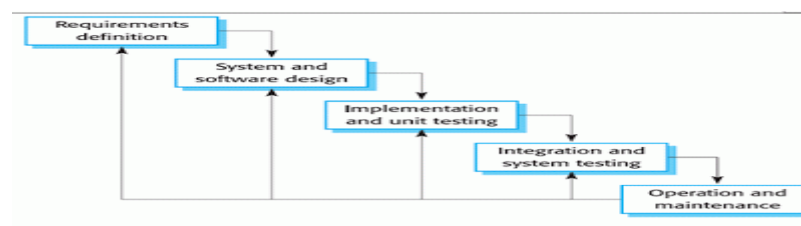
SOA merupakan sebuah solusi yang baik untuk pemodelan sistem di perusahaan atau organisasi besar. Sebab, pemodelan ini memiliki banyak kelebihan, diantaranya(Erl dalam Miharjaya Kardy (2013)) :

1. Dapat menyatukan berbagai sistem yang memiliki platform berbeda, seperti J2EE dan .NET. sebab dengan pendekatan ini, pengembang akan memilih untuk membangun sebuah layer di atas sistem-sistem tersebut yang dapat saling berkomunikasi dengan pesan yang sudah distandarisasi, misalnya menggunakan teknologi XML. Dalam sudut pandang SOA, kedua sistem itu masing-masingnya akan dianggap sebagai service.
2. Tahan terhadap perubahan. Perusahaan atau organisasi besar seringkali berubah struktur untuk meningkatkan efisiensi dan

kinerja. Akibatnya, perangkat lunak juga terkena imbas untuk menyesuaikan diri terhadap proses bisnis yang baru. Pemodelan perangkat lunak dengan SOA akan mengurangi usaha untuk modifikasi perangkat lunak tersebut, karena seluruh logic dari sistem sudah terpartisi secara bersih menjadi sekumpulan services, yang hanya perlu dilakukan yaitu menyusun ulang seluruh service tersebut dan jika perlu menambahkan yang baru. Hal ini jelas akan sangat mengurangi biaya daripada membangun sebuah perangkat lunak yang baru.

2.5 Metode Pengembangan Sistem

Model pengembangan software yang diperkenalkan oleh Winston Royce pada tahun 70-an ini yaitu metode *waterfall* merupakan model klasik yang sederhana dengan aliran sistem yang linier — keluaran dari tahap sebelumnya merupakan masukan untuk tahap berikutnya. Pengembangan dengan model ini adalah hasil adaptasi dari pengembangan perangkat keras, karena pada waktu itu belum terdapat metodologi pengembangan perangkat lunak yang lain. Proses pengembangan yang sangat terstruktur ini membuat potensi kerugian akibat kesalahan pada proses sebelumnya sangat besar dan acap kali mahal karena membengkaknya biaya pengembangan ulang (Angga Setiawan (2013)).



Gambar 2.3 Metode *waterfall* (Angga Setiawan (2013))

Metode *Waterfall* adalah suatu proses pengembangan perangkat lunak berurutan, di mana kemajuan dipandang sebagai terus mengalir ke bawah (seperti air terjun) melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi), dan pengujian (Angga Setiawan (2013)).

Dalam pengembangannya metode *waterfall* memiliki beberapa tahapan yang runtut: *requirement* (analisis kebutuhan), *design* sistem (*system design*), *Coding & Testing*, Penerapan Program, pemeliharaan (Angga Setiawan (2013)).

- *Requirement* (analisis kebutuhan).

Dalam langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau study literatur. Seseorang system analisis akan menggali informasi sebanyak-banyaknya dari user sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh user tersebut. Tahapan ini akan menghasilkan dokumen user requirement atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan sistem. Dokumen inilah yang akan menjadi acuan system analisis untuk menterjemahkan kedalam bahasa pemrograman.

- *Design System* (design sistem)

Proses design akan menterjemahkan syarat kebutuhan sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat koding. Proses ini berfokus pada : struktur data, arsitektur perangkat lunak, representasi interface, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut software requirement.

Dokumen inilah yang akan digunakan programmer untuk melakukan aktivitas pembuatan sistemnya.

- *Coding & Testing* (penulisan sinkode program / implemmentation)

Coding merupakan penerjemahan design dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh programmer yang akan menerjemahkan transaksi yang diminta oleh user. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan computer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan testing terhadap sistem yang telah dibuat tadi. Tujuan testing adalah menemukan kesalahan-kesalahan terhadap system tersebut dan kemudian bisa diperbaiki.

- Penerapan / Pengujian Program (*Integration & Testing*)

Tahapan ini bisa dikatakan final dalam pembuatan sebuah sistem. Setelah melakukan analisa, design dan pengkodean maka sistem yang sudah jadi akan digunakan oleh user.

- Pemeliharaan (*Operation & Maintenance*)

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (periferal atau system operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional.

2.6 Unified Modeling Language (UML)

Secara umum *Unified Modeling Language* (UML) merupakan “bahasa” untuk visualisasi, spesifikasi, konstruksi, serta dokumentasi. Dalam kerangka visualisasi, para pengembang menggunakan UML sebagai suatu cara untuk mengkomunikasikan idenya kepada para pemrogram serta calon pengguna sistem/perangkat lunak. Dengan adanya “bahasa” yang bersifat standar, komunikasi perancang dengan pemrogram (lebih tepat lagi komunikasi antar anggota kelompok pengembang) serta calon pengguna diharapkan menjadi mulus (Nugroho, 2005).

Dalam kerangka spesifikasi, UML menyediakan model-model yang tepat, tidak ambigu, serta lengkap. Secara khusus, UML menspesifikasikan langkah-langkah penting dalam pengambilan keputusan analisis, perancangan, serta implementasi dalam sistem yang sangat bernuansa perangkat lunak (*software intensive system*). Dalam hal ini, UML bukanlah merupakan bahasa pemrograman tetapi model-model yang tercipta berhubungan langsung dengan berbagai macam bahasa pemrograman, sehingga adalah mungkin melakukan pemetaan (*mapping*) langsung dari model-model yang dibuat dengan UML ke bahasa-bahasa pemrograman berorientasi objek (Nugroho, 2005).

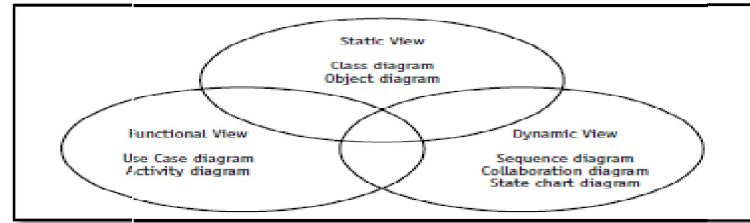
Secara umum UML diterapkan dalam pengembangan sistem/perangkat lunak berorientasi objek sebab metodologi UML ini umumnya memiliki keunggulan-keunggulan dibawah ini (Nugroho, 2005):

1. *Uniformity*. Dengan metodologi UML (atau metodologi berorientasi objek pada umumnya), para pengembang cukup menggunakan 1

metodologi dari tahap analisis hingga perancangan. Hal ini tidak bisa dilakukan dalam metodologi pengembangan terstruktur. Dengan perkembangan masa kini ke arah aplikasi GUI (*Graphical User Interface*), UML juga memungkinkan kita merancang komponen antarmuka pengguna (*User Interface*) secara terintegrasi bersama dengan perancangan perangkat lunak sekaligus dengan perancangan basis data.

2. *Understandability*. Dengan metodologi ini kode yang dihasilkan dapat diorganisasi kedalam kelas-kelas yang berhubungan dengan masalah sesungguhnya sehingga lebih mudah dipahami siapa pun juga.
3. *Stability*. Kode program yang dihasilkan relatif stabil sepanjang waktu sebab sangat mendekati permasalahan sesungguhnya di lapangan.
4. *Reusability*. Dengan metodologi berorientasi objek, dimungkinkan penggunaan ulang kode, sehingga pada gilirannya akan sangat mempercepat waktu pengembangan perangkat lunak (sistem informasi).

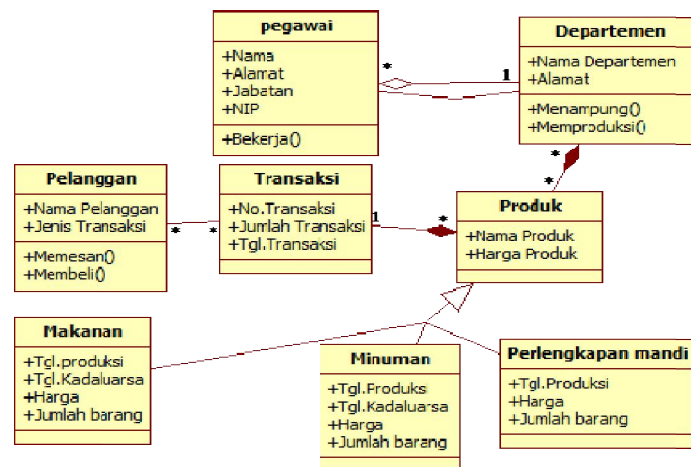
UML terdiri atas pengelompokan diagram-diagram sistem. Diagram adalah yang menggambarkan permasalahan maupun solusi dari permasalahan suatu model. Salah satu cara untuk mengatur diagram UML adalah dengan menggunakan *view*. *View* adalah kumpulan dari diagram yang menggambarkan aspek yang sama dari proyek yang terdiri dari *Static View*, *Dinamis View*, dan *Fungsional View* (Pender, 2002). Gambar 2.4 menggambarkan sifat komplementer dari tiga pandangan dan diagram yang membuat setiap tampilan.



Gambar 2.4 : *Three complementary views*
or sets of diagrams (Pender, 2002)

1. Class Diagram

Bersifat statis, diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif (Nugroho, 2005).



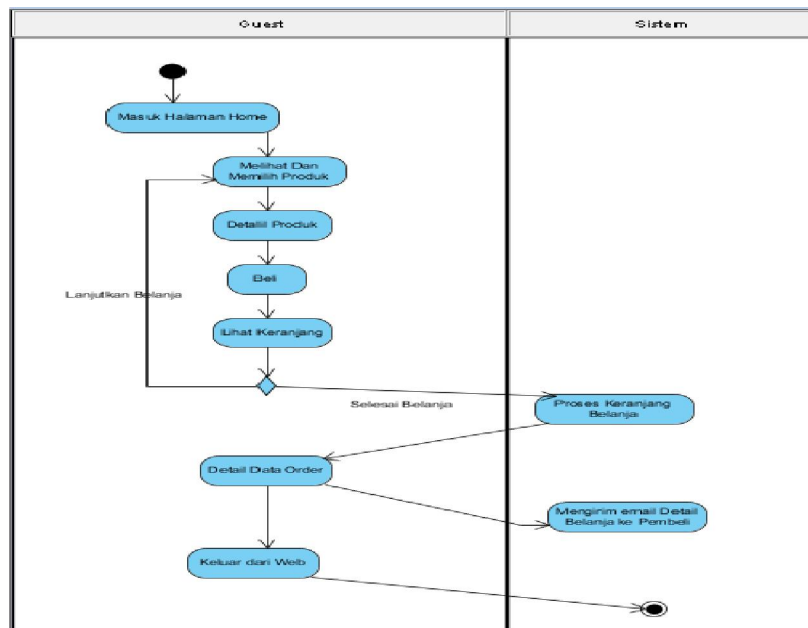
Gambar 2.5: *Class Diagram* (uml-diagrams.org, 2012)

Pada gambar 2.5 *Class / table departemen* memiliki ber-Agregasi dengan *class / table pegawai*, alasannya karena departemen dapat berdiri sendiri tanpa ada pegawai tetapi kinerjanya tidak sempurna. Banyak pegawai dapat bekerja pada satu departemen jadi many to 1. *Class/table transaksi* tidak dapat berdiri sendiri tanpa adanya *class/table produk*.

Begitu juga dengan *table* produk tidak bisa berdiri sendiri tanpa adanya departemen. Banyak pelanggan dapat melakukan banyak transaksi. 1 transaksi dapat mencakup banyak produk.

2. Activity Diagram

Bersifat dinamis, diagram ini adalah tipe khusus dari diagram state yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek (Nugroho, 2005).



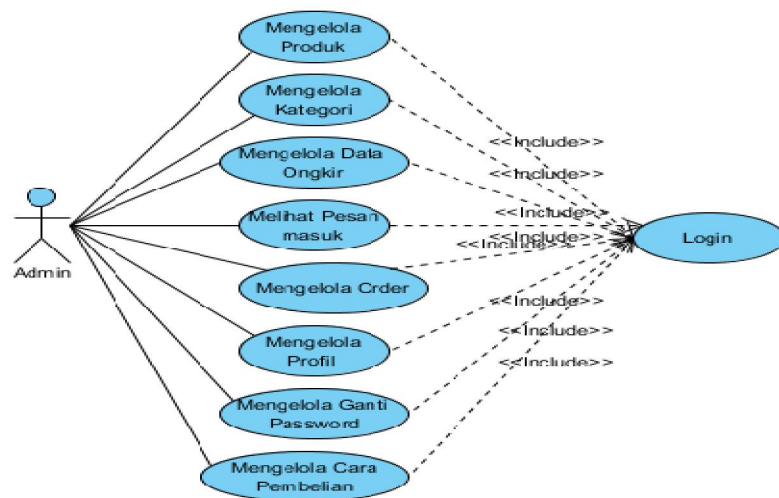
Gambar 2.6 Activity Diagram (uml-diagrams.org, 2012)

Gambar 2.6 menjelaskan sebuah proses pembelian barang. Pembeli masuk ke halaman home, kemudian pembeli dapat melihat dan memilih produk yang akan dibeli. Sebelum menentukan barang yang akan dibeli, pembeli terlebih dahulu melihat detail barang yang akan dipilih. Setelah menentukan barang yang dibeli, pembeli melihat kembali barang apa

saja yang sudah dipilih, jika masih ada yang kurang pembeli dapat kembali memilih barang, jika tidak ada barang yang ingin ditambah maka system akan melakukan proses keranjang dan system akan memperlihatkan detail barang-barang yang dibeli oleh pembeli dan mengirimkan email detail belanja kepada pembeli. Setelah semuanya selesai maka pembeli dapat meninggalkan atau keluar website.

3. *Usecase Diagram*

Bersifat statis, diagram ini memperlihatkan himpunan *usecase* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna (Nugroho, 2005).

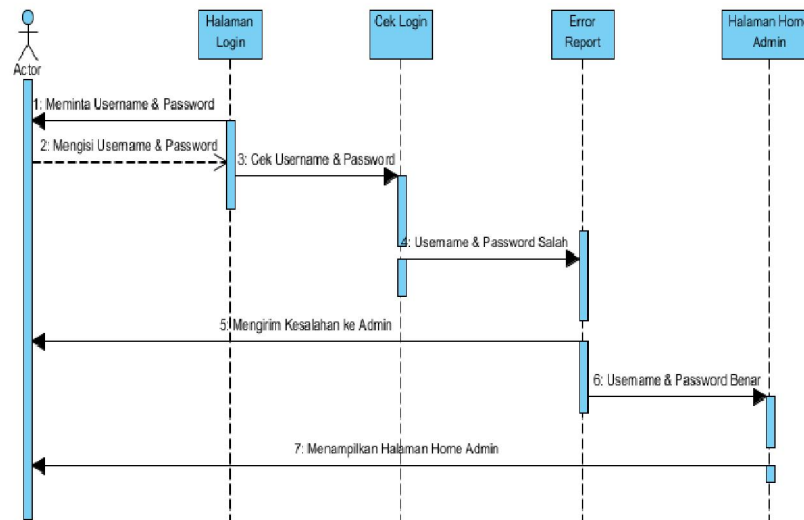


Gambar 2.7: *Usecase Diagram* (uml-diagrams.org, 2012)

Gambar 2.7 memperlihatkan sebuah proses yang dapat dilakukan oleh admin. Untuk dapat melakukan proses yang ada seperti gambar di atas, admin terlebih dahulu melakukan login.

4. Sequence Diagram

Bersifat dinamis, diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan (*message*) dalam suatu waktu tertentu (Nugroho, 2005).



Gambar 2.8: Sequence Diagram (uml-diagrams.org, 2012)

Gambar 2.8 memperlihatkan sebuah proses yang dapat dilakukan oleh Actor. Pertama *Actor* masuk ke halaman login. Sistem akan meminta *username* dan *password* actor tersebut, kemudian *actor* memasukkan *username* dan *password*. Selanjutnya *system* akan melakukan cek terhadap *username* dan *password* yang telah dimasukkan. Jika *username* dan *password* salah, maka akan tampil *report error* dan *system* akan mengirimkan kesalahan data yang dimasukkan oleh actor. Kemudian, setelah *actor* melihat kesalahan data yang dimasukkan, maka actor memasukkan kembali *username* dan *password* yang benar. Jika *username* dan *password* yang dimasukkan sudah benar, maka *system* akan menampilkan halaman utama.

2.7 Pengujian Sistem

Pengujian pada dasarnya adalah menemukan serta menghilangkan 'bug' (kesalahan-kesalahan) yang ada di sistem/perangkat lunak. Kesalahan-kesalahan itu dapat diakibatkan beberapa hal utama, antara lain kesalahan saat penentuan spesifikasi sistem, kesalahan saat melakukan analisis permasalahan, kesalahan saat perancangan, serta kesalahan saat implementasi (Nugroho, 2005).

2.7.1 Teknik Pengujian Sistem

Konsep kualitas sangat penting demi kepuasan pengguna (juga pengembang). Untuk mencapai kualitas yang diharapkan dari sistem yang kita kembangkan pada umumnya ada beberapa strategi pengujian yang dapat dilakukan. Strategi-strategi itu adalah (Bahrawi, 1999 dalam Nugroho, 2005: 435-436):

1. *Black-Box Testing*. Pada pengujian ini kita tidak perlu tahu apa sesungguhnya terjadi pada sistem/perangkat lunak. Yang kita uji adalah masukkan serta keluarannya. Artinya, dengan berbagai masukkan yang kita berikan, apakah sistem memberikan keluaran seperti yang kita harapkan?
2. *White-Box Testing*. Pengujian jenis ini mengasumsikan bahwa spesifikasi logika adalah penting dan perlu dilakukan pengujian untuk menjamin apakah sistem berfungsi dengan baik. Tujuan utama dari strategi pengujian ini adalah pengujian berbasis kesalahan.

3. *Top-DownTesting*. Pengujian ini berasumsi bahwa logika utama atau interaksi antar objek perlu diuji lebih lanjut. Strategi ini seringkali dapat mendeteksi cacat/kesalahan/kekurangan yang serius. Pendekatan ini sesuai dengan strategi pengujian berbasis scenario.
4. *Bottom-Up Testing*. Strategi ini mulai dengan rincian sistem kemudian beranjak ke peringkat yang lebih tinggi. Dalam metodologi berorientasi objek, kita mulai dengan menguji metoda-metoda dalam kelas, menguji kelas-kelas serta interaksi antarkelas, dan selanjutnya hingga pada peringkat yang paling tinggi.

2.7.2 Pengukuran Tingkat Kepuasan Pengguna

Mengenai kepuasan pengguna kita masih harus meninjau seberapa jauh sistem yang kita kembangkan memuaskan pengguna. Beberapa cara yang dapat ditempuh untuk mengetahui kepuasan pengguna adalah:

1. **Kuestioner**. Kuestioner merupakan daftar pertanyaan yang diajukan pada seorang responden untuk mencari jawaban dari permasalahan yang diteliti (Hasibuan, 2007).
2. **Pengamatan Langsung**. Pengamatan langsung bisa juga dilakukan untuk mengetahui apakah sistem yang kita kembangkan sesuai dengan kebutuhan serta harapan pengguna.

2.8 Penelitian Terkait

Penelitian terkait dari penelitian yang dilakukan ini adalah :

1. Rancang Bangun Sistem Informasi Minimarket Dengan Penerapan *Service Oriented Architecture*(Miharjaya Kaddy, 2008), Perguruan Tinggi Universitas Siliwangi tasikmalaya. Metode yang digunakan pada aplikasi ini yaitu metode *Service Oriented Architecture* (SOA). SOA dapat merespon lebih cepat dengan biaya yang lebih efektif. Aplikasi ini nantinya berguna untuk mengintegrasikan layanan yang berbeda secara luas untuk lingkungan yang berbasis web dan menggunakan *flatfom* yang berbeda-beda.
2. Implementasi *Service-Oriented Architecture* dengan *Web Service* untuk Aplikasi Informasi Akademik(Kapojas F, 2012) , Perguruan tinggi Fakultas Teknik UNSRAT. Dalam penelitian ini *SOA* diimplementasikan menggunakan teknologi *Web Service* yaitu sebagai aplikasi yang dibuat untuk dapat diakses oleh aplikasi lain dengan menggunakan *XML* sebagai format pengiriman pesan. Dengan adanya penerapan *Service Oriented Architecture* pada penelitian ini diharapkan dapat diintegrasikan dengan aplikasi informasi akademik Fakultas Teknik UNSRAT.
3. Implementasi *Cloud Computing* untuk Memaksimalkan Layanan Pariwisata(Wahyudi Tri, 2013),Perguruan Tinggi AMIK BSI YOGYAKARTA. Aplikasi yang dibuat ini menggunakan cloud computing. Dengan di implementasikannya *Cloud Computing* pada aplikasi pelayanan pariwisata dapat menjadi solusi agar biaya

penerapan teknologi informasi di bidang pariwisata menjadi lebih murah dan dapat memaksimalkan layanan pariwisata.

4. Komputasi Awan (Cloud Computing) Perpustakaan Pertanian(Syaikhu Akhmad, 2013), Pustakawan Muda pada Pusat Perpustakaan dan Penyebaran Teknologi Pertanian (PUSTAKA). Aplikasi Perpustakaan pertanian ini menerapkan *cloud computing*. Tulisan ini bertujuan untuk memberikan gambaran tentang konsep penerapan *cloud computing* di perpustakaan, sehingga di masa yang akan datang perpustakaan sebagai penyedia layanan informasi dapat memberikan layanan yang terbaik, mutakhir dan berkesinambungan kepada penggunanya. Dengan berbekal informasi yang ada, pengguna dapat melakukan berbagai pengkajian, penelitian atau keperluan lain untuk melahirkan pemikiran dan inovasi yang dapat bermanfaat bagi khalayak luas.
5. Rancang Bangun Sistem Persediaan (*Inventory*) Dengan Model *Software As A Service* Menggunakan *Service Oriented Architecture*(Rifai Ahmad, dkk, 2012), Perguruan Tinggi Institut Teknologi Sepuluh November. Aplikasi ini dibangun dengan menggunakan analisis dan desain SOAD. Dan berhasil diimplementasikan menggunakan teknologi Java serta *web service* sebagai komunikasi antar domain fungsi yang dilakukan dalam *enterprise resource*. Aplikasi yang dibuat pada tugas akhir ini mampu mengakomodasi fitur *Inventory* yaitu pembelian material (*purchasing*) dan pengiriman barang (*delivering*). Dari data

transaksi purchasing dan delivering, aplikasi ini dapat memperkirakan jumlah yang optimal suatu barang yang tersimpan di gudang(*controlling stock*). Sehingga tidak terjadi kelebihan stok maupun kekosongan stok

BAB III

METODOLOGI PENELITIAN

3.1 Jenis Penelitian

Jenis Penelitian yang digunakan dalam perancangan simulasi ini termasuk jenis penelitian terapan yaitu penelitian yang dilakukan berkenaan dengan kenyataan-kenyataan praktis, penerapan, dan pengembangan ilmu pengetahuan yang dihasilkan oleh penelitian dasar dalam kehidupan nyata. Tujuan utama penelitian terapan adalah pemecahan masalah sehingga hasil penelitian dapat dimanfaatkan untuk kepentingan manusia baik secara individu atau keperluan industry atau politik dan bukan hanya untuk wawasan keilmuan semata .

Dari pengertian di atas, maka penelitian ini termasuk penelitian terapan, dengan menerapkan Service Oriented Architecture terhadap interkoneksi antara server akademik Universitas Bengkulu dan server dengan Server Bank.

3.2 Sarana Pendukung

Dalam penelitian ini sarana pendukung yang digunakan berupa perangkat keras (*hardware*) dan perangkat lunak (*software*) yang berperan penting dalam hal desain system, *generate code* pada system.

1. Perangkat keras (*hardware*) yang digunakan antara lain :

Minumun :

- Laptop dengan spesifikasi Processor Intel (R) Core (TM) 1.66 Ghz, RAM 1GB, Hard Disc 248 GB.

Optimal :

- Laptop dengan spesifikasi Processor Intel (R) atom (TR) 2.13 Ghz, RAM 1GB, Hard Disc 500 GB.
- Modem

2. Perangkat lunak (software) yang digunakan antara lain :

- Sistem Operasi Windows 7
- Bahasa pemrograman PHP

3.3 Jenis dan Sumber Data

Jenis data yang dibutuhkan dan digunakan dalam penelitian ini adalah data kualitatif. Data kualitatif merupakan jenis data yang dapat diukur secara langsung atau dapat dihitung. Data-data yang dikumpulkan dalam penelitian ini berasal dari 2 (dua) sumber data yaitu:

1. Data Primer

Data Primer yang dibutuhkan berupa format data yang didapatkan dari pihak Bank BNI dan data akademik mahasiswa yang melakukan proses pembayaran mahasiswa baru, registrasi ulang.

2. Data Sekunder

Data sekunder diperoleh dari berbagai sumber yaitu skripsi, laporan, jurnal, dan lain-lain.

3.4 Teknik Pengumpulan Data

Dalam mengumpulkan data, teknik pengumpulan data yang digunakan dalam penelitian ini yaitu :

a. Artikel

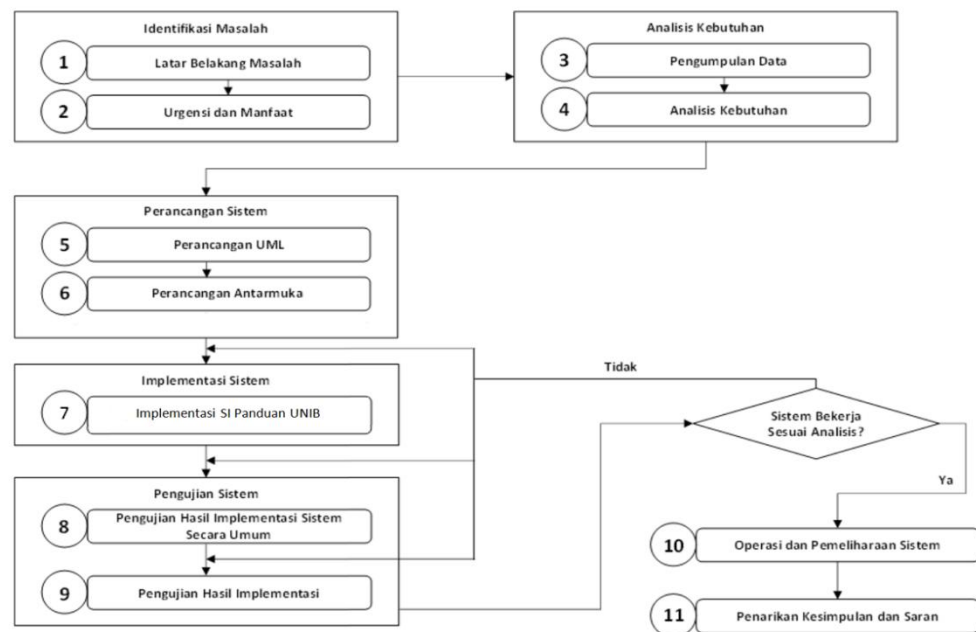
Artikel yang digunakan sebagai sumber data dalam penelitian ini adalah artikel yang didapat dari internet yang berhubungan dengan penelitian ini.

b. Buku

Buku yang digunakan dalam penelitian ini berupa buku-buku referensi yang dapat menunjang materi mengenai bahasa pemrograman PHP dan buku-buku yang membahas mengenai service oriented architecture.

3.5 Metode Pengembangan Sistem

Penelitian dilakukan berdasarkan diagram alir dibawah ini, hal ini juga disesuaikan dengan metode pengembangan sistem yang dipilih. Adapun diagram alir pada penelitian ini dapat digambarkan sebagai berikut:



Gambar 3.1 Diagram Alir Penelitian

Penjelasan dari diagram alir penelitian diatas adalah sebagai berikut:

1. Pada tahap pertama adalah penentuan basis awal dari sebuah penelitian yaitu latar belakang penelitian.
2. Tahap kedua yaitu menentukan tujuan, manfaat dan ruang lingkup penelitian. Target pencapaian dalam tahap ini adalah diketahuinya tujuan dan manfaat dari aplikasi informasi Universitas Bengkulu sebagai panduan pengenalan kampus berbasis android. Sedangkan batasan masalah digunakan untuk membatasi pembahasan dan ruang lingkup penelitian agar tidak terlalu luas.
3. Pada tahap ketiga dilakukan pengumpulan data-data yang diperlukan dalam penelitian yang dibutuhkan dalam membangun aplikasi informasi Universitas Bengkulu sebagai panduan pengenalan kampus berbasis android. Pengumpulan data tersebut akan diperoleh dari studi kepustakaan, tracking titik koordinat dari sarana kampus dan wawancara di Lembaga Pendidikan dan Evaluasi Universitas Bengkulu.
4. Tahap keempat adalah proses analisis kebutuhan. Analisis kebutuhan yang ditentukan oleh peneliti terdiri dari analisis kebutuhan sistem, analisis kebutuhan proses, analisis kebutuhan input, analisis kebutuhan output, dan analisis kebutuhan perangkat lunak dan perangkat keras.
5. Tahap kelima yaitu perancangan Unified Modelling Language (UML). Pada tahapan ini akan diketahui semua entitas luar, input, dan output yang terlibat dalam sistem serta diagram use-case,

diagram kelas, diagram aktifitas, diagram objek, diagram state chart, diagram kolaborasi, dan diagram sequence yang digunakan dalam analisis sistem.

6. Tahap keenam yaitu perancangan flowchart dan antarmuka (human interface). Tahap perancangan flowchart akan digunakan untuk menggambarkan sistem baru yang akan dikembangkan secara logis tanpa mempertimbangkan terlebih dahulu lingkungan sistem. Tahap perancangan antarmuka akan dibuat dalam beberapa rancangan tata letak system sesuai dengan analisis kebutuhan dari sistem.
7. Tahapan ketujuh yaitu implementasi sistem secara keseluruhan. Implementasi berdasarkan hasil dari tahapan keempat hingga ketujuh, yaitu dari tahapan analisis kebutuhan, UML, flowchart dan antarmuka.
8. Tahapan kedelapan yaitu tahapan pengujian hasil implementasi sistem secara umum. Pengujian meliputi pengujian white box dan pengujian black box.
9. Tahapan kesembilan yaitu tahap pengujian simulasi pemanfaatan php xml sebagai webservice dengan menggunakan teknologi cloud dengan menggunakan metode *Service Oriented Architecture* untuk meningkatkan kinerja sistem registrasi akademik Unib.
10. Tahapan kesepuluh adalah operasi dan pemeliharaan sistem. Tahapan ini dilakukan setelah proses pengujian sistem telah

berhasil dilakukan. Pengoperasian dilakukan langsung oleh pengguna sistem yaitu mahasiswa dan masyarakat

11. Tahapan terakhir adalah penarikan kesimpulan dan saran. Tahap tersebut dilakukan untuk mengetahui hasil yang telah diperoleh selama melakukan penelitian.

3.6 Metode Uji Kelayakan Sistem

Uji Kelayakan dilakukan untuk mendapatkan penilaian langsung terhadap sistem yang dihasilkan. Target dari pengujian kelayakan sistem ini adalah responden (calon pemakai sistem). Adapun tahapan dari uji kelayakan ini adalah:

1. Kuisisioner

Kuisisioner atau angket merupakan daftar pertanyaan yang diajukan pada seorang responden untuk mencari jawaban dari permasalahan yang diteliti. Teknik pemilihan responden (sampel) dilakukan dengan metode *Stratified sampling*. *Stratified sampling* yaitu pengambilan sampel dengan cara bertingkat dan biasanya digunakan oleh peneliti apabila di dalam populasi terdapat strata atau tingkatan antara satu kelompok dengan kelompok lainnya (Mulyatiningsih, 2012).

2. Tabulasi Data.

Proses perhitungan data angket menggunakan skala *likert*. Banyak orang yang bingung jika menggunakan *Skala Likert* dan bahkan salah pemahaman. *Skala Likert* digunakan untuk membuat angket,

tapi kadang-kadang salah isi yang disasar untuk dihimpun dengan *Skala Likert* tersebut. *Likert* itu nama orang, lengkapnya *Rensis Likert*, pendidik dan ahli psikologi Amerika Serikat. Jadi, skala ini digagas oleh *Rensis Likert*, sehingga disebut *Skala Likert*. *Skala likert* adalah perhitungan skor pada tiap-tiap *interval* dari pernyataan yang diberikan ke responden. Skala itu sendiri salah satu artinya, sekedar memudahkan, adalah ukuran-ukuran berjenjang. Skala penilaian, misalnya, merupakan skala untuk menilai sesuatu yang pilihannya berjenjang, misalnya 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. *Skala Likert* juga merupakan alat untuk mengukur (mengumpulkan data dengan cara “mengukur-menimbang”) yang “itemnya” (butir-butir pertanyaannya) berisikan (memuat) pilihan yang berjenjang. Hasil dari proses perhitungan disajikan dalam bentuk tabel. Sehingga didapatkan nilai uji kelayakan terhadap sistem. Untuk apa sebenarnya *Skala Likert* itu? *Skala Likert* itu “aslinya” untuk mengukur kesetujuan dan ketidaksetujuan seseorang terhadap sesuatu objek, yang jenjangnya bisa tersusun atas:

1. Sangat setuju
2. Setuju
3. Netral antara setuju dan tidak
4. Kurang setuju
5. Sama sekali tidak setuju

Sebagai contoh perhitungan dengan menggunakan *skala likert* yaitu pada uraian kasus berikut ini :

- Berikan jawaban pertanyaan berikut sesuai dengan pendapat anda, dengan cara memberikan tanda (√) pada kolom yang tersedia.

No.	Pertanyaan	Jawaban				
		SS	ST	RG	TS	STS
1	Prosedur kerja yang baru itu akan segera diterapkan di perusahaan anda,		√			
2					

SS = Sangat Setuju diberi skor 5

ST = Setuju diberi skor 4

RG = Ragu-ragu diberi skor 3

TS = Tidak Setuju diberi skor 2

STS = Sangat Tidak Setuju diberi skor 1

Kemudian dengan teknik pengumpulan data angket, maka instrument tersebut misalnya diberikan kepada 100 orang karyawan yang diambil secara random. Dari 100 orang karyawan setelah dilakukan analisis misalnya:

25 Orang menjawab SS

40 Orang menjawab ST

5 Orang menjawab RG

20 Orang menjawab TS

10 Orang menjawab STS

Berdasarkan data tersebut 65 orang (40+25) atau 65% karyawan menjawab setuju dan sangat setuju. Jadi kesimpulannya mayoritas karyawan setuju dengan adanya metode kerja baru.

Data interval tersebut juga dapat dianalisis dengan menghitung rata-rata jawaban berdasarkan skoring setiap jawaban dari responden. Berdasarkan skor yang telah ditetapkan dapat dihitung sebagai berikut:

Jumlah skor untuk 25 orang yang menjawab SS = $25 \times 5 = 125$

Jumlah skor untuk 40 orang yang menjawab ST = $40 \times 4 = 160$

Jumlah skor untuk 5 orang yang menjawab RG = $5 \times 3 = 15$

Jumlah skor untuk 20 orang yang menjawab TS = $20 \times 2 = 40$

Jumlah skor untuk 10 orang yang menjawab STS = $10 \times 1 = 10$

Jumlah Total = 350

Jumlah skor ideal (kriterium) untuk seluruh item = $5 \times 100 = 500$ (seandainya semua menjawab SS). Jumlah skor yang diperoleh dari penelitian = 350. Jadi berdasarkan data itu maka tingkat persetujuan terhadap metode kerja baru itu = $(350 : 500) \times 100\% = 70\%$ dari yang diharapkan (100%).

3.7 Jadwal Penelitian

Berikut ini adalah jadwal penelitian yang dilaksanakan:

Tabel 3.1 Jadwal Penelitian yang Akan Dilaksanakan

No	Kegiatan	Bulan					
		Jan 2014	Feb 2014	Mar 2014	Apr 2014	Mei 2014	Juni 2014
1.	Pengajuan Judul Proposal						
2.	Penyusunan Proposal						
3.	Seminar Proposal						
4.	Perbaikan Proposal						
5.	Pengumpulan Data						
6.	Analisis dan Perancangan Sistem						
7.	Pembuatan Coding						
8.	Implementasi dan Pengujian Unit						
9.	Integrasi dan Pengujian						
10.	Analisis Hasil						
11.	Sidang Hasil						